

TER : Bras Robot pour jeu de plateau

Berriri Mehdi, Bounab Fayçal, Djema Sofiane, Doglio Arthur

June 2019

Sommaire

I/	Introduction	2
1/	Présentation du groupe	2
2/	Présentation du sujet	2
II/	État de l'art	3
1/	Contexte	3
2/	Technologies utilisées	3
III/	Travail effectué	5
1/	Études	6
2/	Algorithme	12
IV/	Gestion de projet	14
1/	Organisation	14
2/	Github	15
V/	Conclusion	16
1/	Tache restantes	16
2/	Problèmes rencontrés	16
VI/	Perspectives et réflexions personnelles	16

I/ Introduction

Dans le cadre de notre de notre Master 1 Informatique nous avons souhaité prendre comme travail d'étude et de recherche le sujet qui porte le nom "Bras Robot pour jeu de plateau". Nous avons choisi ce sujet parce qu'on pense que les robots sont un excellent moyen pour expérimenter l'apprentissage par les essais et les erreurs .

1/ Présentation du groupe

Notre groupe est composé de : Berriri Mehdi, Bounab Fayçal, Djema Sofiane, Doglio Arthur et nos intervenants sont : Mr Renevier Philippe et Mme Pelleau Marie .

2/ Présentation du sujet

Le but de ce TER est de développer une librairie de commande du bras de robot Joy-It ROBOT02, ou améliorer une librairie existante et de mettre en oeuvre quelques mouvements dans un jeu réel, en fixant les emplacements ou en utilisant une librairie de reconnais-

sance pour les éléments d'un jeu.

II/ État de l'art

1/ Contexte

Dans ce TER nous devions au début travailler sur le jeu du Takenoko mais nous voulions un autre jeu plus pratique pour le bras robotique, nous avons donc choisi le jeu d'échecs. Notre but est que notre bras robotique puisse jouer quelques coups qu'on lui donne face à un humain, des coups précis.

2/ Technologies utilisées

Nous utilisons l'environnement de développement d'Arduino, le langage de programmation utilisé est le C++, compilé avec `avr-g++ 9`, et lié à la bibliothèque de développement Arduino, permettant l'utilisation de la carte et de ses entrées/sorties. La mise en place de ce langage standard rend aisé le développement de programmes sur les plates-formes Arduino à toute personne maîtrisant le C ou le C++.



FIGURE 1 – Arduino IDE

Nous utilisons une carte Arduino qui est la base de la programmation avec microcontrôleur (un « mini-ordinateur » qui se présente sous la forme d'un circuit intégré). Il existe de nombreuses versions de cette carte. Actuellement nous avons une Arduino Uno wifi rev2



FIGURE 2 – Arduino Uno Wifi REV2

Nous avons un bras robot Joy-it ROBOT02



FIGURE 3 – Bras Joy-it ROBOT02

III/ Travail effectué

Au tout début du Ter nous avons commencé à travailler sur une Arduino Leonardo étant donnée que nous somme plus à l'aise avec la programmation en c/c++ . Après s'être bien familiarisé (à peu près 6 semaines) avec le bras robotique nous avons entamé un travail en

parallèle sur une raspberry Pi 3 pour implémenter le Wi-Fi car nos encadrants nous ont donné la contrainte de pour contrôler le bras à distance. Malheureusement ce fut un échec dû à notre manque de temps et notre méconnaissance de la programmation sur Raspberry. Nous avons fait quelques recherches sur différents sites nous avons trouvé l'Arduino, une Uno Wi-Fi REV2 qu'il nous fallait et nous avons demandé à nos encadrants de nous la procurer

1/ Études

a/ Arduino

Arduino Leonardo

La Leonardo est sortie après la carte Arduino Uno, et a été dotée d'un nouveau type de processeur, et d'une ergonomie revue. Elle a le même nombre de ports que la carte Arduino Uno (12 analogiques et 20 numériques dont 7 pwm). Son gros plus : un port USB, qui vous permettra par exemple d'émuler un clavier ou une souris ! Cette carte sera très bien pour débiter un projet de domotique, comme la gestion de l'éclairage de sa maison ou la fermeture/ouverture des volets en fonction de

la météo extérieure. Vous pouvez l'utiliser bien entendu sur des projets de niveau intermédiaire.

Arduino Uno Wifi REV2

(14 broches d'entrée/sortie numériques (dont 6 peuvent être utilisées comme sorties PWM), 6 entrées analogiques, un résonateur céramique de 16 MHz, une connexion USB, une prise d'alimentation, un en-tête ICSP et un bouton de réinitialisation. Elle comporte tout ce qui est nécessaire à la prise en charge du microcontrôleur. Il vous suffit de la connecter à un ordinateur à l'aide d'un cordon USB ou de l'alimenter avec un adaptateur c.a./c.c. ou une batterie. Le module Wi-Fi est un SoC autonome avec pile de protocole TCP/IP qui peut donner accès à votre réseau Wi-Fi. (Ou l'appareil peut agir en tant que point d'accès). Une caractéristique utile de Uno Wi-Fi est la prise en charge de la programmation OTA (over-the-air), que ce soit pour le transfert d'esquisses Arduino ou de firmware Wi-Fi. Il comprend un nouveau microprocesseur 8 bits à partir de Microchip Il est doté d'un IMU (unité de mesure d'inertie) Vous pouvez vous connecter à votre réseau Wi-Fi avec le module wifi intégré La connexion wifi est sécurisée par le nouvel accélérateur de puce cryptée ECC608)

b/ Raspberry

Nous avons essayé d'utiliser une raspberry pi 3 à la place de l'arduino mais nous avons eu des difficultés pour la configuration de celle-ci, de plus notre code actuel est en C++ et pour la raspberry c'est du Python. Il faut aussi un écran pour bosser directement sur la raspberry ou créer une VM afin de voir l'interface de la raspberry qui a un environnement Unix

c/ Bras Robotique

Nous avons fais une étude sur les différents bras robotiques avec lesquels nous pourrions travailler certains sont plus performant mais le prix est plus élevé.

Bras robotique Joy-It ROBOT02

Les encadrants nous ont imposé le robot JOY-IT ROBOT02, ce bras robuste livré avec servomoteurs, pince et visserie de montage. Le bras est équipé de 6 puissants servomoteurs de déplacement, 3 pince (rotation, inclinaison et ouverture/fermeture) et 3 pour le bras (coude, épaule et rotation de la base). Pour un coût de 176,40

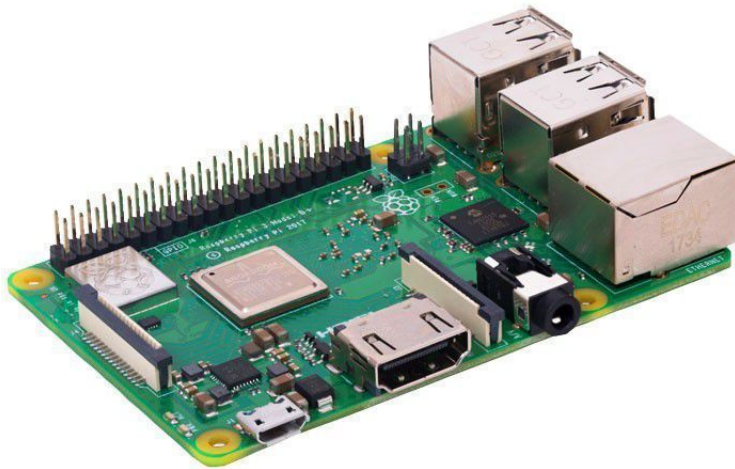


FIGURE 4 – Raspberry PI 3

euros

Bras robotisé en kit FreaksArm EF08070

Bras motorisé didactique commandé via une carte Freaduino compatible Arduino à assembler soi-même avec châssis en bois, servomoteurs permettant des mouvements sur 4 axes et 2 joysticks. Pour un coût de 75,90

euros

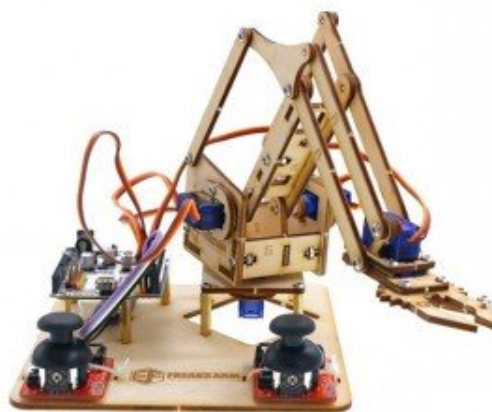


FIGURE 5 – Bras FreaksArm

Bras Dobot Basic

Le bras robotisé et modulaire Dobot Basic, destiné au milieu éducatif, permet de s'initier à la robotique, à la programmation et à la conception 3D. Ce bras accepte de nombreuses d'applications comme la réalisation de dessins, l'impression 3D et le déplacement d'objets.

Les éléments permettant d'obtenir toutes ces fonctionnalités sont inclus dans le kit et sont facilement intégrables au bras (vis et cordon inclus). Le châssis est en aluminium anodisé et en plastique ABS assurant une

bonne robustesse.

Le conception de ce bras permet d'obtenir des opérations avec une précision de 0,2 mm grâce à ses 4 axes et ses différents moteurs. Il peut être contrôlé de plusieurs façons via un PC par cordon USB, par bluetooth (smartphone et tablette, application Android et iOS disponible) et peut être programmé en C++, C, Python et Java. Pour un coût de 1215 euros.



FIGURE 6 – Bras Dobot Basic

Bras robotique Braccio T050000

Le bras motorisé et modulaire Braccio d'Arduino est livré en kit à assembler soi-même et se contrôle via une carte Arduino ou compatible (non incluse). Ce bras peut effectuer différents mouvements grâce à un ensemble de

servomoteurs et dispose d'une pince pouvant saisir des objets jusqu'à 150 g. Le bras Braccio est polyvalent et sa modularité lui permet d'obtenir différentes configurations. Pour un coût de 238 euros.



FIGURE 7 – Bras Braccio T050000

2/ Algorithme

Notre bras est composé de 6 cerveaux (moteurs); deux d'entre eux font des rotations (le premier cerveau qui sert de plate-forme et le cinquième qui fait tourner la pince), un autre permet l'ouverture et la fermeture de la pince, et les derniers permettent la prise de la pièce

d'échec. Ce bras évoluera dans un environnement 3D (x, y, z).

a/ Calibrage

Dans cette étape nous avons démonté chacun des cerveaux du robot afin de fixer ses positions initiales et son orientation .

b/ Calcul

Avant de commencer à calculer quoi que ce soit nous avons pris des multiples mesures pour trouver les différentes distance entre les différents éléments du robots (distance entre les moteurs) ainsi que les limites et les équivalences d'angles entre le réel et le code (disponible dans le document Angles sur github).

Nous sommes ensuite passé à la partie "mathématique" du TER c'est-à-dire calculer les différents angles de chacune des articulations afin d atteindre l'objectif. Voici quelque représentation graphique du schéma général

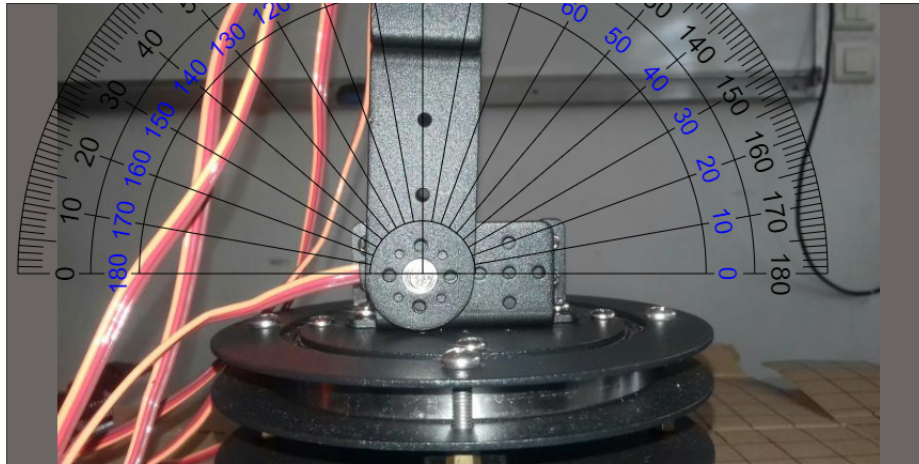


FIGURE 8 – Prise de mesure

c/ Le code

Dans le code il y a une méthode de calcul des angles en prenant en compte des coordonnées (x, y, z) , une méthode de conversion (degrés en radian) et une méthode de déplacement des articulations.

IV/ Gestion de projet

1/ Organisation

Niveau organisation on se voyait de façon hebdomadaire en fin de semaine aux campus des lucioles afin de travailler ensemble et discuter de l'avancement du Ter et si notre objectif hebdomadaire n'était pas atteint, nous continuions chez nous.

Nous organisons aussi des rendez-vous avec nos encadrants toutes les trois semaines dans le but de leur montrer ce qu'on a fait et ce qu'on voudra faire pour les semaines à venir (On a mis un journal de bord sur notre Github)

(Il faut rajouter les tâches de chaque personne dans le projet)

2/ Github

Dans le but de mieux s'organiser et d'avoir une trace écrite des difficultés rencontrées nous avons mis en place un dépôt sur GitHub.

Notre dépôt est composé de deux parties : une partie "Docs" qui contient différents fichiers (angles, étude, journal de bord) , et une partie "code" (les classes et les méthodes qui font bouger le bras).

Voici le lien du Github :

<https://github.com/faycalbounab/TER-Bras-Robot>

v/ Conclusion

1/ Tache restantes

2/ Problèmes rencontrés

VI/ Perspectives et réflexions personnelles

Pour aller plus loin dans ce projet, nous pourrions utiliser des capteurs afin de développer une IA pour pouvoir jouer sans à avoir à donner d'instructions aux bras dans une partie d'échecs.